Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Previously Presented) An apparatus comprising:

a prefetch engine to prefetch data from a distributed, coherent memory in response to a first transaction from an input/output bus directed to the distributed, coherent memory; and

an input/output coherent cache buffer to receive the prefetched data, the coherent cache buffer being coherent with the distributed, coherent memory and with other cache memories in a system including the input/output coherent cache buffer,

the prefetch engine further to speculatively prefetch data in anticipation of a need for the speculatively prefetched data in association with a second input/output transaction if data has been prefetched for pending, memory-related transactions from the input/output bus.

2. (Original) The apparatus of claim 1 wherein the prefetch operation performed by the prefetch engine is a non-binding prefetch operation such that the prefetched data received by the coherent cache buffer may be altered by a memory in the distributed coherent memory.

3. (Original) The apparatus of claim 2 wherein the first transaction request is a memory read request and the prefetch engine issues a read request to prefetch data to be read from the distributed, coherent memory in response to the first transaction request.

4. (Original) The apparatus of claim 2 wherein the first transaction request is a memory write request and the prefetch engine issues a request to prefetch ownership of a memory line in the distributed, coherent memory, the memory line being indicated by the first transaction request.

5. (Original) The apparatus of claim 1 further comprising:

an input/output transaction request buffer to temporarily store transaction requests received from the input/output bus directed to the distributed, coherent memory.

6. (Original) The apparatus of claim 5 wherein

the prefetch engine prefetches data in response to transaction requests stored in the input/output transaction request buffer.

7. (Original) The apparatus of claim 6 wherein

the prefetch engine prefetches data in response to transaction requests stored in the input/output transaction request buffer regardless of the order in which the transaction requests were received from the input/output bus.

8. (Original) The apparatus of claim 5 further comprising:

a retire engine to retire input/output transaction requests stored in the transaction request buffer in program order after the transaction requests have been completed.

9. (Original) The apparatus of claim 8 wherein

the retire engine is further to check the input/output coherent cache buffer to determine whether data associated with an input/output transaction request to be retired is present in the input/output coherent cache buffer in a valid state.

10. (Original) The apparatus of claim 1 wherein

coherency is maintained between the input/output coherent cache buffer and the distributed, coherent memory using a MESI protocol.

11. (Previously Presented) A method comprising:

prefetching data in response to a first input/output transaction request received from an input/output bus and directed to a distributed, coherent memory;

if data has been prefetched for pending memory-related input/output transactions, speculatively prefetching data in anticipation of a need for the speculatively prefetched data in association with a second input/output transaction;

temporarily storing the prefetched data; and maintaining coherency between the prefetched data and data stored in the distributed, coherent memory and data stored in other cache memories.

12. (Original) The method of claim 11 further comprising:

buffering input/output transaction requests received from the input/output bus that are directed to the distributed, coherent memory.

13. (Previously Presented) The method of claim 12 further comprising:

prefetching data in response to third and fourth buffered input/output transactions wherein

prefetching data in response to the first, third and fourth buffered input/output transactions may be performed in any order.

14. (Original) The method of claim 12 further comprising:

retiring the buffered input/output transactions in the order in which they were issued by the input/output bus.

15. (Original) The method of claim 14 wherein retiring includes

checking the temporarily stored, prefetched data to determine whether valid data corresponding to the transaction request to be retired is temporarily stored.

16. (Original) The method of claim 11 wherein

maintaining coherency includes maintaining coherency using a MESI protocol.

17. (Original) The method of claim 11 wherein prefetching includes

issuing a request for the data in response to the first transaction request; and

receiving the requested data.

18. (Previously Presented) The method of claim 17 wherein

prefetching data in response to a third input/output transaction request received from the input/output bus and directed to the distributed, coherent memory occurs between issuing the request and receiving the requested data.

19. (Previously Presented) A computer system comprising:

first and second processing nodes each including at least one processor and at least one caching agent;

a distributed coherent memory wherein portions of the distributed coherent memory are included within each of the first and second processing nodes; and

an input/output node coupled to the first and second processing nodes, the input/output node comprising

a prefetch engine to prefetch data from the distributed, coherent memory in response to a first transaction from a first input/output bus directed to the distributed, coherent memory and to speculatively prefetch data after data has been prefetched for pending memory-related transactions from the input/output bus in anticipation of a need for the speculatively prefetched data in association with a second input/output transaction; and

an input/output coherent cache buffer to receive the prefetched data, the coherent cache buffer being coherent with the distributed, coherent memory and the caching agents.

20. (Original) The computer system of claim 19 further comprising:

a coherent system interconnect to couple each of the first and second processing nodes to the input/output node, the coherent system interconnect to communicate information to maintain coherency of the distributed, coherent memory and to maintain coherency between the input/output coherent cache buffer and the distributed, coherent memory.

21. (Original) The computer system of claim 20 wherein coherency is maintained in accordance with a MESI protocol.

22. (Original) The computer system of claim 19 further comprising

an interconnection network to communicate information between the first and second processing nodes and the input/output node.

23. (Original) The computer system of claim 19 further comprising

an input/output bridge coupled between the first and second processing nodes and a plurality of input/output buses, the plurality of input/output buses including the first input/output bus, the input/output bridge including the prefetch engine and the input/output coherent cache buffer.

24. (Previously Presented) The computer system of claim 23 wherein the input/output bridge further comprises:

at least one input/output transaction request buffer to temporarily store input/output transaction requests received from the plurality of input/output buses that are directed to the distributed, coherent memory.

25. (Original) The computer system of claim 24 wherein

the prefetch engine prefetches data in response to transaction requests stored in the input/output transaction request buffer regardless of the order in which the transaction requests are stored.

26. (Previously Presented) The computer system of claim 24 wherein the input/output bridge further comprises

a retire engine further to check the input/output coherent cache buffer for valid data corresponding to a transaction request to be retired,

the retire engine to retire transaction requests stored in the input/output transaction request buffer in program order.